

Bots incluidos

- *dummy*: blanco inmóvil
- *rabbit*: se mueve mucho pero no dispara
- *rook*: dificultad baja
- *counter*: dificultad media
- *grunt*: dificultad alta

*- only goes in ONE direction until it reaches border
or
res another direction*

Control del Tanque

Recuerda que el tanque...

- ...*sigue conduciendo* hasta que le digamos lo contrario
- ...*tiene inercia*: tarda en acelerar y en frenar
- ...*no gira a más de 50km/h*

Control del Tanque

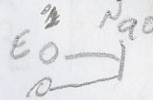
El tanque tiene un *radar* para explorar la arena

- direccional, con una apertura máxima de ± 10 grados

Arena

La arena es un rectángulo de 1340x1000

- El origen (0,0) es la esquina inferior izquierda
- El ángulo 0 es el **Este** y el ángulo 90 es el **Norte**



Control del Tanque

El tanque tiene 100 puntos de blindaje

- Es *destruido* si su blindaje baja *por debajo* de 0

- Controlamos el tanque mediante los métodos de *tank*

```
async function main(tank) {  
  while (true) {  
    // tu código aquí  
    await tank.drive(0, 50);  
  }  
}
```

Control del Tanque

El blindaje recibe daño cuando:

- Le alcanza la explosión de un misil
- El tanque se choca contra la pared de la arena
- El tanque se choca contra otro tanque

Control del Tanque

Tu tanque puede *disparar misiles*

- En cualquier dirección
- Solo puede haber *dos misiles* en el aire al mismo tiempo
- Tienen un *alcance máximo* de 700 metros
- Se pueden programar para explotar en un rango determinado

Entorno de Desarrollo

¿Cómo programo mi IA?

En *robot.js*:

- Declara una función *main* (punto de entrada)
- Recibe un parámetro *tank* ✓
 - el objeto con el que vamos a controlar nuestro robot
- Controlamos el tanque mediante los métodos de *tank*

in CONFIG.SS

Programar tu IA

- Puedes utilizar tantas variables como quieras ✓
- Pero siempre tienen que estar declaradas dentro de "main"

Programar tu IA

Para definir nuevas funciones respeta tres reglas:

- Tiene que estar dentro de *main*
- Tienes que definirla siempre con *async* delante ✓
- Tienes que invocarla siempre con *await* delante ✓

call

Programar tu IA

- Asegúrate de que *main* no termina ✓
- Escribe tu código dentro de un bucle *while (true)* ✓

Programar tu IA

Para clasificarse para el torneo, tu IA tiene que...


- Ser capaz de matar a *rabbit* en un uno contra uno
- ¡En 60 segundos o menos!

22/50
6-5-21
Ete di-1
Shi!


```

async function main(tank) {
  while (true) {
    // tu código aquí
    await tank.drive(0, 50);
  }
}

```



```

async function main(tank) {
  // variables aquí
  let lastPosition = 0;

  // punto de entrada
  while (true) {
    lastPosition = await tank.getX();
    console.log(lastPosition);
  }
}

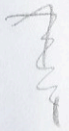
```

Tegan

```

async function main(tank) {
  // definicion de otras funciones
  async function stop(tank) {
    while (await tank.getSpeed() > 50) {
      await tank.drive(0, 0);
    }
  }
}

```



```

// punto de entrada
while (true) {
  await tank.drive(0, 100);
  while (await tank.getX() < 800) { }
  await stop(tank);
  await tank.drive(180, 100);
  while (await tank.getX() > 200) { }
  await stop(tank);
}
}

```


Control del Tanque ✓

await tank.drive(angle, speed)

- **angle**: dirección (0-359)
- **speed**: velocidad (0-100)

Control del Tanque

Consultamos la posición del tanque con dos métodos:

- await tank.getX();
- await tank.getY();

Control del Tanque

elejira
stop 3so gira
Para girar primero tenemos que frenar hasta 50km/h

- Podemos consultar la velocidad del tanque
- await tank.getSpeed();

Control del Tanque ✓

await tank.scan(angle, resolution);

- **angle**: dirección (0-359)
- **resolution**: apertura del cono (1-10)
- si hay *uno o más tanques* en el cono:
 - devuelve distancia *al más cercano*
- si *no hay tanques* devuelve 0

Control del Tanque ✓

await tank.shoot(angle, range);

- **angle**: dirección (0-359)
- **range**: alcance (1-700)

Control del Tanque ✓

Podemos consultar el estado de nuestro blindaje:

- await tank.getDamage()
- devuelve el daño recibido
- blindaje restante = 100 - daño